

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau



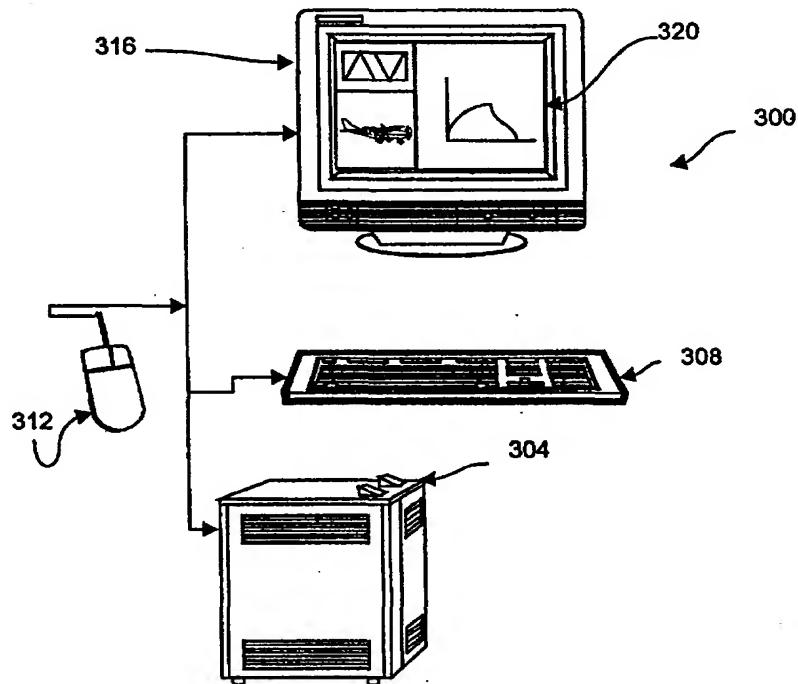
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁷ : G06F 3/033	A1	(11) International Publication Number: WO 00/63768 (43) International Publication Date: 26 October 2000 (26.10.00)
(21) International Application Number: PCT/CA00/00393		(81) Designated States: AU, CA, JP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).
(22) International Filing Date: 17 April 2000 (17.04.00)		
(30) Priority Data: 09/292,967 16 April 1999 (16.04.99) US		Published <i>With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>
(71) Applicant: AVID TECHNOLOGY, INC. [US/US]; One Park West, Tewksbury, MA 01876 (US).		
(71)(72) Applicant and Inventor: SHEASBY, Michael, C. [CA/CA]; 3510 Boul. St-Laurent, Montreal, Quebec H2X 2V2 (CA).		
(72) Inventor: ROUSSEAU, Luc-Eric; Apartment #1110, 3475 Rue St-Urbain, Montreal, Quebec H2X 2N4 (CA).		
(74) Agents: GRAHAM, Robert, J. et al.; Gowling Lafleur Henderson LLP, Suite 4900, Commerce Court West, Toronto, Ontario M5L 1J3 (CA).		

(54) Title: SYSTEM AND METHOD FOR MANAGING THE USE OF DESKTOP SPACE IN GRAPHICAL USER INTERFACES

(57) Abstract

A system and method for managing the use of the desktop in a graphical user interface system includes trays, which are manager windows that can manage other windows. Space on the desktop can be designated a tray and each tray can be populated by one or more other windows, including other trays. Multiply populated trays provides a tray switcher tool by which a user can select for display any one of the windows managed by the trays. While it is preferred to define trays as flat windows, which do not move front to back, on the desktop, trays can be defined, as floating windows, where desired.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

SYSTEM AND METHOD FOR MANAGING THE USE OF DESKTOP SPACE IN GRAPHICAL USER INTERFACES

FIELD OF THE INVENTION

5 The present invention relates to graphical user interfaces. More specifically, the present invention relates to a system and method of managing the use of desktop space in graphical user interfaces and navigating the user interface.

BACKGROUND OF THE INVENTION

10 Graphical user interfaces (GUIs) are widely used in many computer operating system shells and application programs. Most GUIs have common features including resizable and repositionable windows which are displayed to users on a desktop, which is a display area on one or more computer monitors. While GUIs have proven to be a significant advantage in improving the ease of use and usefulness of many programs, they do suffer from some
15 disadvantages.

One such disadvantage is that application programs which offer rich and diverse toolsets and components for performing tasks often have a great many user interface objects which must be displayed for these toolsets and components. The desktop area for such programs quickly becomes very crowded and, in fact, it is often not possible or practical to display all of
20 the desired user interface objects on the desktop at a single time. An example of an application program with a rich set of user interface objects is the SOFTIMAGE|DS digital post production program sold by the assignee of the present invention. SOFTIMAGE|DS has many modes of operation which offer a diverse set of toolbars, viewers, browsers and other user interface objects which are displayed to the user.

25 As it is often not possible or practical to display all of their user interface objects on the desktop at one time, many programs change the displayed set of user interface objects according to the mode of the application program. For example, the SOFTIMAGE|DS program has several modes, each of which has a defined layout of user interface components and these layouts include an Editing layout, a Compositing layout, a Graphics layout, and a Media Management layout. Each layout defines an arrangement including the displayed user interface objects and their screen positions, for a corresponding mode. In addition, in SOFTIMAGE|DS
30 layouts can be created and/or modified by users to suit their particular preferences and/or needs.

The provision of user-modifiable layouts also assists a user in obtaining an interface which is more intuitive, for the user, to navigate. Otherwise, as is often the case, a user must make a significant effort to learn to navigate within a program with rich sets of user interface objects that have been organized and arranged according to the design of the authors
5 of the program.

Despite the use of layouts and the changing of user interface objects displayed for each mode of a program, programs with rich GUIs like SOFTIMAGE|DS generally also require the user of a layout to switch their attention between various windows, some of which float over the desktop and/or each other. In the above-mentioned SOFTIMAGE|DS program, the user interface comprises both "flat" windows that are positioned on the desktop and which do not move to the front of the display to obscure other windows, and "floating" windows that are displayed in front of the flat windows. User interface objects which are required much of the time are displayed in the flat windows and user interface components needed intermittently, or less often, are displayed in floating windows.
10

15 For example, in the Edit Layout of the SOFTIMAGE|DS program, Clip Browser, Image Viewer, Transport Control and Timeline components are displayed as flat windows and components such as the Animation Editor and a Property Editor are displayed, as needed, in floating windows. When a user activates one of these floating windows, for example by clicking in it with a pointing device, it is moved to the front of other windows in the display so that the
20 user has an unobstructed view of the activated window. However, when a floating window is moved to the front of the display, it obscures the portion of any windows, floating or flat, that are behind it.

Such systems of floating and flat windows suffer from disadvantages in that a user
25 must still manage which window or windows (floating or flat) are obscured and which windows are at the front of the desktop. In systems with many such windows and/or in which the user needs to switch between windows very often, this management task can be onerous. Typically a user spends significant amounts of time and effort to size and position floating windows on the desktop in an attempt to obtain an arrangement that serves the workstyle of the user and their preferences. These arrangements are often created so that selecting a floating window to perform
30 a task will not obscure another window that the user believes is related to that task. For example, the floating window for the above-mentioned Property Editor would likely be positioned such that it does not obscure the flat window for the Image Viewer.

Prior attempts to manage graphical user interface have employed two broad classes of implementations, namely: independent windows; and managed systems of split windows. An example of independent windows is Microsoft Word 97. Here, multiple windows representing different open documents can be displayed simultaneously and it is up to the user to manage their relative positions onscreen. It is time-consuming to manually size and position these windows such that they tile (i.e. - abut adjacent windows) neatly against one another in an effort to avoid having windows pop up and occlude one another during normal workflow. While automated tiling is available, which automatically arranges the windows such that they lie neatly beside one another, the arrangements are 'rigid', in that the positions are determined by the number of open windows, and not by user preference, and 'unpredictable', in that the user cannot easily predict which window will be placed where and thus cannot become accustomed to seeing a particular window in the same position each time they run the program.

An example of a managed system of split windows is Microsoft's Visual Studio Version 5. Here, the desktop is comprised of a set of vertical and horizontal divider bars that separate multiple windows. These divider bars can be moved, to resize adjacent windows. Windows can be moved from one desktop area to another, or to float above the divided desktop. One disadvantage of this system is that the number of windows that the user can concurrently work with is limited to the total flat area of the desktop, as the windows must appear side by side. In addition, even when floating windows are also employed by users in an attempt to address this limited desktop area, the floating windows themselves are subject to the same independent windows limitations, discussed above.

While such previous attempts at managing the use of desktop space in a graphical user interface have been workable, they still require a considerable effort on the part of the user to manage the arrangements of windows as changes between modes of the application program occur and/or as additional, possibly less frequently used, user interface objects are required to be displayed.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide a novel system and method for managing the use of desktop space in graphical user interfaces which obviates or mitigates at least some of the above-identified disadvantages of the prior art.

According to a first aspect of the present invention, there is provided a method of managing the use of desktop space in a graphical user interface, comprising the steps of:

- (i) defining a region to be a manager window;
- (ii) populating said manager window with at least one window to be managed thereby;
- (iii) sizing each window populating said manager window to be displayed in the area of said defined region;
- (iv) selecting and displaying one of said windows populating said manager window in said defined region; and
- (v) upon receiving input from a user representing the selection of a different one of said windows populating said managed window, replacing the display of said selected one window in said defined region with the display of said different one of said windows populating said manager window.

According to another aspect of the present invention, there is provided a graphical user interface system for managing the use of desktop space, comprising:

- a processor;
- user input means connected to said processor to provide input from a user to said processor; and
- user output means connected to said processor to receive output from said processor in response to said provided input and to display to a user a desktop of a graphical user interface incorporating said received output in at least one window, said graphical user interface including at least one manager window which manages said at least one window.

The present invention provides a system and method for managing the use of space on the desktop in a graphical user interface system. Space on the desktop can be designated as a tray, which is a manager window that can manage other windows. Trays can be multiply populated by windows or other trays in two ways: by sharing desktop area multiple windows separated by splitter bars, or swapping desktop area a 'tray switcher' tool. The tray switcher tool allows a user to select for display in the tray any one of the windows managed by the tray. As a window which populates a tray can itself be a tray, sophisticated but intuitive and convenient user interfaces can be constructed. While it is preferred to define trays as flat windows, trays can also be defined as floating windows, where desired.

BRIEF DESCRIPTION OF THE DRAWINGS

Preferred embodiments of the present invention will now be described, by way of example only, with reference to the attached Figures, wherein:

5 Figure 1 shows a schematic representation of a rich graphical user interface desktop including flat windows;

Figure 2 shows the graphical user interface of Figure 1 wherein floating windows have also been employed;

10 Figure 3 shows a schematic representation of a graphical user interface desktop with a pair of trays in accordance with an embodiment of the present invention;

Figure 4 shows the desktop of Figure 3 wherein another tray has been added;

15 Figure 5 shows the desktop of Figure 4 wherein two of the trays have been resized;

Figure 6 shows the desktop of Figure 5 wherein one of the trays has been removed and the two remaining trays have been resized;

20 Figure 7 shows a schematic representation of a sketch for a desired arrangement of trays;

Figure 8 shows the desktop arrangement of trays resulting from a best fit operation performed for the sketch of Figure 7;

25 Figure 9 shows the desktop of Figure 4 wherein each tray has been populated with a single window;

Figure 10 shows the desktop of Figure 9 wherein one tray has been populated with two windows;

Figure 11 shows the desktop of Figure 10 wherein the window displayed in one tray has been changed;

25 Figure 12 shows the desktop of Figure 10 wherein each tray has been populated with multiple windows;

Figure 13a shows a tray with multiple windows;

Figure 13b shows the tray of Figure 13a wherein one of the windows in the tray set is a tray having two singly populated trays, separated by a splitter bar;

30 Figure 13c shows the tray of Figure 13a wherein one of the windows in the tray set is a tray populated with a multiply populated tray and a singly populated tray;

Figure 14 shows the desktop of Figure 12 further including a floating tray which is multiply populated; and

Figure 15 shows a graphical user interface system in accordance with the present invention.

5

DETAILED DESCRIPTION OF THE INVENTION

A graphical user interface desktop for a digital post production editing system is indicated generally at 20 in Figure 1. Desktop 20 shows a layout suitable for editing and which includes a Clip Browser component 24, an Image Viewer component 28, a Transport Control component 32 and a Timeline component 36. These components also can have one or more toolbar components 40 associated with them. In the editing layout of desktop 20, each of the above-mentioned components is displayed as a "flat" window which does not move to the front of the desktop when active. These components are selected for display in flat windows, based upon the assumption by the author of the layout that these components are those most often referred to by a user when in editing mode.

Figure 2 shows desktop 20 when three floating windows 44, 48 and 52 have been opened by the user to access functions or tools not available in the layout of displayed flat windows. As shown, each floating window obscures parts of the flat windows on the desktop displayed behind it and, in fact, floating window 48, which is the front window, also obscures parts of floating windows 44 and 52 which are behind it.

Figure 3 shows a desktop 100 in accordance with an embodiment of the present invention. In the present invention, any unoccupied space in desktop 100 is defined as a manager window, referred to herein as a "tray", which is a window that manages other windows, as described below. A user can subdivide any tray to create two or more additional trays, by positioning a splitter bar, or bars, in the tray to geometrically define the desired additional trays. In the example of Figure 3, the desktop was defined as tray 108 and the user has positioned a splitter bar 104 across tray 108 to create a new tray 112.

Positioning a splitter bar can be accomplished in any suitable manner, as will occur to those of skill in the art. For example, in a present embodiment of the invention, a user accesses a contextual pop up menu for a tray that it is desired to split by, for example, right clicking on the tray with a pointing device such as a mouse. The appropriate selection is made from the available splitting choices ("split horizontally", "split vertically", "split two ways", or

10

15

20

25

30

"split four ways") on the pop up menu and the corresponding splitter bar or bars are displayed and employed to define the new additional trays. In Figure 3, the user selected "split vertically" from a pop up menu displayed for tray 108, to add splitter bar 104 and create tray 112.

Splitting a tray four ways produces a matrix of nine separate trays, a configuration which, as will be appreciated by one skilled in the art, cannot be achieved with any combination of horizontal and vertical splits. With this selection, the existing tray becomes the center tray of the resulting matrix of new trays.

Figure 4 shows desktop 100 after a user has horizontally split tray 112 to create new tray 116, which is separated from tray 112 by splitter bar 120. In a present embodiment of the invention, when a tray is split into additional trays, the splitter bar or bars are, by default, initially arranged according to where the user next clicks after making the desired choice from the contextual menu. The user is guided in his choice by an XOR'd line that is displayed in the GUI to indicate where the splitter bar will be positioned when the user next clicks in the display.

A user can also resize trays by moving the appropriate splitter bars by a suitable user interface operation, for example by left-dragging the bars (dragging the mouse while holding down the left mouse button), to obtain the desired relative sizes.

Figure 5 shows desktop 100 wherein a user has moved splitter bar 120 to reduce the display size of tray 116 and to correspondingly increase the display size of tray 112.

Also, a tray can be removed by selecting the tray to be removed on desktop 100 and indicating that it is to be removed, by any suitable means. For example, a user can right click on a flat window tray to be removed from desktop 100 to access a pop up contextual menu from which the user can select 'remove tray'. The most recent splitter bar touching this tray will be removed, and the view on the opposite side of that splitter bar will be resized to occupy the sum of its display space and the display space of the tray which was removed. The 'most recent' splitter bar is predictable given that trays are always subdivided in a progressive manner, which provides an unambiguous ordering to splitter bar creation. In the example shown in the configuration of Figures 5 and 6, the user has selected tray 108 and chosen to remove it. The most recent splitter bar for tray 108 is splitter bar 104, which is removed. As illustrated in this example, both of trays 112 and 116 are resized horizontally to occupy the display space formerly occupied by removed tray 108.

A tray in a floating tray that contains splitter bars can be removed by selecting the 'remove tray' item on a pop up contextual window for the floating tray. Alternatively the entire

floating tray can be removed from a layout by simply closing it using a 'close' button built into the frame of all floating windows. As a floating tray is not defined by splitter bars, no reuse is made of the area it occupied after it is removed, although the area of the desktop or any other floating windows it obscured will now be visible.

5 To reduce the time and effort required for a user to design a layout of trays, the present invention can also allow a user to "sketch" and "best fit" a tray layout on desktop 100. In sketch mode, which the user can select by a suitable user interface event, the user can roughly size and position on desktop 100 a set of user interface components, or placeholder components as described below, to be arranged in separate trays.

10 Figure 7 shows an example of such a sketch wherein user interface components 140, 144, 148, 152, 156 and 160 have been roughly sized and positioned on desktop 100. Once the sketch is complete, the user indicates, again by any suitable user interface event, that it is desired to exit sketch mode and that a best fit is desired for the sketch. A resulting desktop for the set of trays, as shown in Figure 8, is then produced from the sketch.

15 If the user has employed user interface components, such as a Clip Browser component, Image View component, etc. in the sketch, these components are correspondingly placed into the resulting trays. If the user has employed one or more placeholders components in the sketch, the corresponding trays in the resulting tray layout are empty, i.e. - have not been populated with interface components. As will be apparent to those of skill in the art, a user can 20 adjust the resulting best fit tray layout, if desired, by manually repositioning splitter bars, as described above.

25 When a tray is created, or at anytime thereafter, a tray can be populated with one or more user interface components (i.e. - windows) to form a tray set. A tray set comprises a collection of windows that are managed by the tray, along with their arrangement within that tray, for which the tray set is defined.

30 A tray set can be created, or modified, by dragging and dropping or otherwise indicating that a window is to be added to the set. Floating windows can be placed into a tray by an appropriate user interface mechanism, such as clicking a button on the surface of the window which drops it into the tray directly beneath that button. Windows already populating a tray can be dragged into other trays, for example by dragging them directly, when in an Edit Tray Layout mode, or by dragging the icon representing that window in a tray switcher, described below, to another tray. Windows can also be directly populated into a tray, for

example by right-clicking on a tray in Edit Tray Layout mode and selecting the desired window from a list of available windows.

Figure 9 shows the desktop of Figure 4, wherein a Clip Browser component has been added to the tray set of tray 108, a Timeline component has been added to the tray set of tray 112 and Image Viewer component has been added to the tray set of tray 116.

Figure 10 shows desktop 100 of Figure 9 when an Animation Editor component has been added to the tray set of tray 112. Once a tray set is populated with more than one window, a tray switcher 180 is displayed within the tray. Tray switcher 180 displays a user interface item, either a pictorial or text icon, for each window in the tray set. In Figure 10, icon 184 corresponds to the Timeline component of the tray set of tray 112 and icon 188 corresponds to the Animation Editor component of that tray set. By selecting one of icons 184 or 188, by any suitable user interface event, the corresponding window will be displayed in tray 112. For example, selecting icon 184 will result in desktop 100 being updated to show the Timeline component in frame 112, as shown in Figure 11. Each tray allows a user to position the tray switcher 180 at the top, bottom, left or right edges of the tray as desired.

Figure 12 shows the desktop 100 of Figure 10 wherein each tray 108, 112 and 116 has been populated with multiple windows and accordingly provided with a tray switcher 180. As will be apparent to those of skill in the art, when a tray is resized, by repositioning one or more splitter bars, each window managed by the tray is resized appropriately, whether the window is presently displayed or not.

In another embodiment of the present invention, the display of tray switcher 180 in a multiply populated tray can be omitted. The user can change the window displayed in a tray by other means, including an appropriate key (e.g. pressing the 'a' key brings the Animation Editor window into view) or key sequence. Each time the tray is selected with a defined user interface event, the corresponding window of the tray is displayed. In this manner, a user can change the displayed contents of a tray, without requiring the use of any portion of the display area of the tray for the task switcher 180.

Figure 13a shows a tray 200 which is populated with three windows, as indicated by the three icons in tray switcher 180. The present invention allows any tray, including those in a tray set, to be split into two or more trays, as shown in Figure 13b, wherein one of the windows of tray 200 has been split into singly populated trays 204 and 208. Further, the present

invention allows a tray in a tray set to be multiply populated, as shown in Figure 13c, wherein tray 204 has been populated with two windows and now includes a tray switcher 180.

While presently preferred for use with flat windows, the instant invention is not so limited and can be employed with floating windows.

5

Figure 14 shows a desktop 100 wherein a floating window comprising a multiply populated tray 212 is employed.

In Figure 15, a graphical user interface system in accordance with the present invention is indicated generally at 300. System 300 includes a computer processor 304, which can be a general purpose processor such as an IBM PC compatible processor with an Intel Pentium II processor, running an operating system such as Microsoft Windows NT. Processor 304 can also be a special purpose processor, such as a process control system for example.

Processor 304 receives user input from user input means, which can include a keyboard 308, a pointing device 312 such as a mouse or graphics tablet, or from any other suitable user input device, as will occur to those of skill in the art. Processor 304 is also connected to a user output means, such as a computer monitor 316 which can be a CRT, LCD or plasma display, or any other suitable output means as will occur to those of skill in the art.

In use, processor 304 receives input from a user, via the user input means, and/or from an operating system or one or more application programs executing on processor 304. In response to this input, processor 304 updates the graphical user interface displayed to the user on the user output means, such as monitor 316. The graphical user interface 320.

The present invention provides a system and method for managing the use of the desktop area in a graphical user interface system. Space on the desktop can be designated a tray and each tray can be populated by one or more other windows, including other trays. Multiply populated trays can provide a tray switcher tool, or can have defined keys or key sequences, by which a user can select for display any one of the windows managed by the trays. While it is preferred to define trays as flat windows, trays can be defined as floating windows, where desired. Trays provide a simple and intuitive method and system for making effective use of the desktop space in a graphical user interface.

The above-described embodiments of the invention are intended to be examples of the present invention and alterations and modifications may be effected thereto, by those of skill in the art, without departing from the scope of the invention which is defined solely by the claims appended hereto.

30

WE CLAIM:

1. A method of managing the use of desktop space in a graphical user interface, comprising the steps of:

- 5 (i) defining a region to be a manager window;
- (ii) populating said manager window with at least one window to be managed thereby;
- (iii) sizing each window populating said manager window to be displayed in the area of said defined region;
- 10 (iv) selecting and displaying one of said windows populating said manager window in said defined region; and
- (v) upon receiving input from a user representing the selection of a different one of said windows populating said manager window, replacing the display of said selected one window in said defined region with the display of said different one of said windows populating 15 said manager window.

2. The method of claim 1 wherein at least one said windows populating said manager window is a manager window.

20 3. The method of claim 1 where in step (v) said received input comprises a user selection of an icon representing said different one of said windows populating said manager window from a set of icons displayed in said manager window, each icon in said set corresponding to a different one of said windows populating said manager window.

25 4. The method of claim 1 wherein at least one additional manager window can be defined by the steps of:

- 30 (a) selecting an existing manager window;
- (b) indicating at least one of a desired horizontal split, vertical split, two-way split and four-way split of said existing manager window to define a desired set of subregions;

- (c) dividing said defined region in accordance with said indicated at least one split and resizing and repositioning said existing manager window into one of said desired subregions; and
- (d) for each remaining desired subregion creating a new manager window.

5

5. The method of claim 1 wherein a desired user interface component window can be added to said windows populating said manager window by selecting said desired user interface component from a menu of user interface component windows displayed by said manager window.

10

6. The method of claim 1 wherein a desired user interface component window can be added to said windows populating said manager window by selecting, dragging and dropping said desired user interface component window onto said manager window within said graphical user interface.

15

7. The method of claim 1 wherein said defined region is on said desktop and said manager window is a flat window.

20

8. The method of claim 7 wherein a manager window can be removed from said desktop by the steps of

- (a) selecting an existing manager window to be removed;
- (b) determining the most recent splitter bar touching said existing manager window;
- (c) removing said indicated manager window from said desktop and resizing the manager windows which abutted the opposite side of said splitter bar to also occupy the defined region of said removed manager window.

25

9. The method of claim 1 wherein said manager window is displayed as a floating window on said desktop.

30

10. The method of claim 1, further comprising the steps of:

- (a) sizing and positioning at least two user interface component windows on said desktop;
- (b) determining a defined region of said desktop for each of said at least two user interface component windows in accordance said sizing and positioning; and
- (c) for each of said at least two user interface component windows, performing steps (i) through (v) to create corresponding manager windows being populated with said respective user interface component window in the corresponding determined defined region for said user interface component window.

5

10

11. A graphical user interface system for managing the use of desktop space, comprising:
a processor;
user input means connected to said processor to provide input from a user to said
processor; and

user output means connected to said processor to receive output from said
processor in response to said provided input and to display to a user a desktop of a graphical user
interface incorporating said received output in at least one window, said graphical user interface
including at least one manager window which manages said at least one window.

15
20

1/9

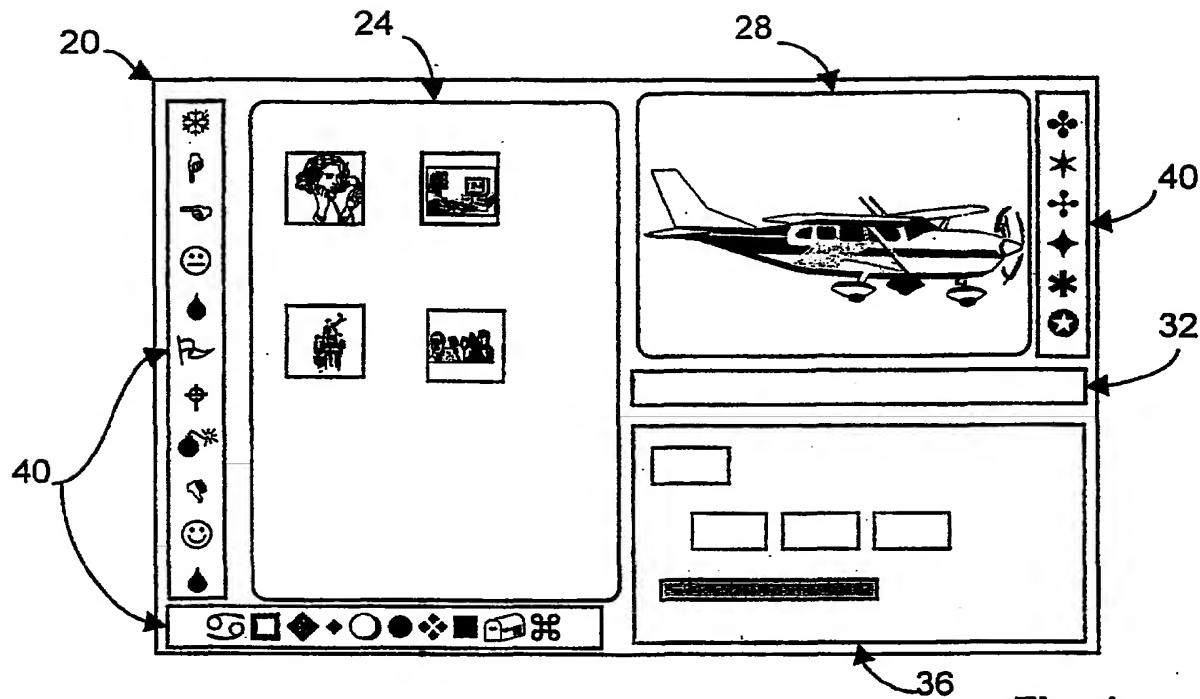


Fig. 1
(prior art)

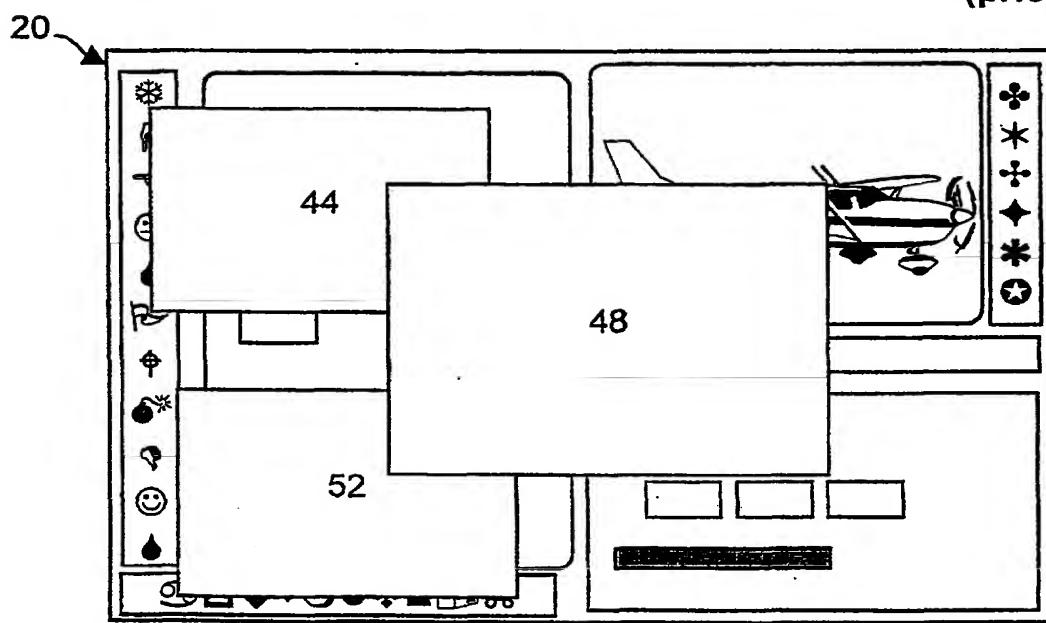


Fig. 2
(prior art)

2/9

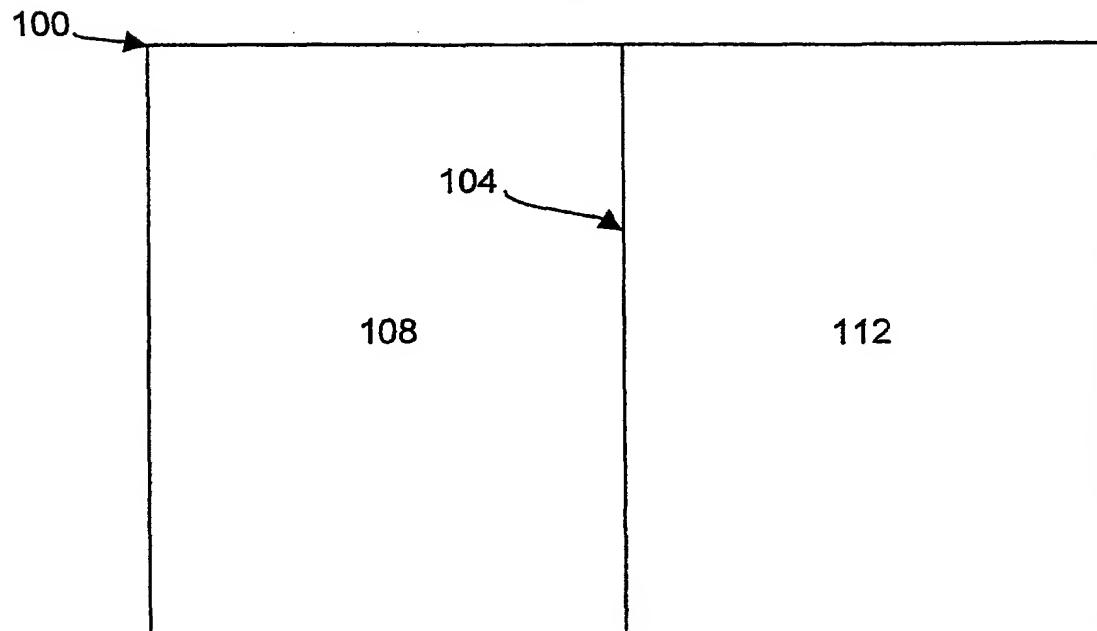


Fig. 3

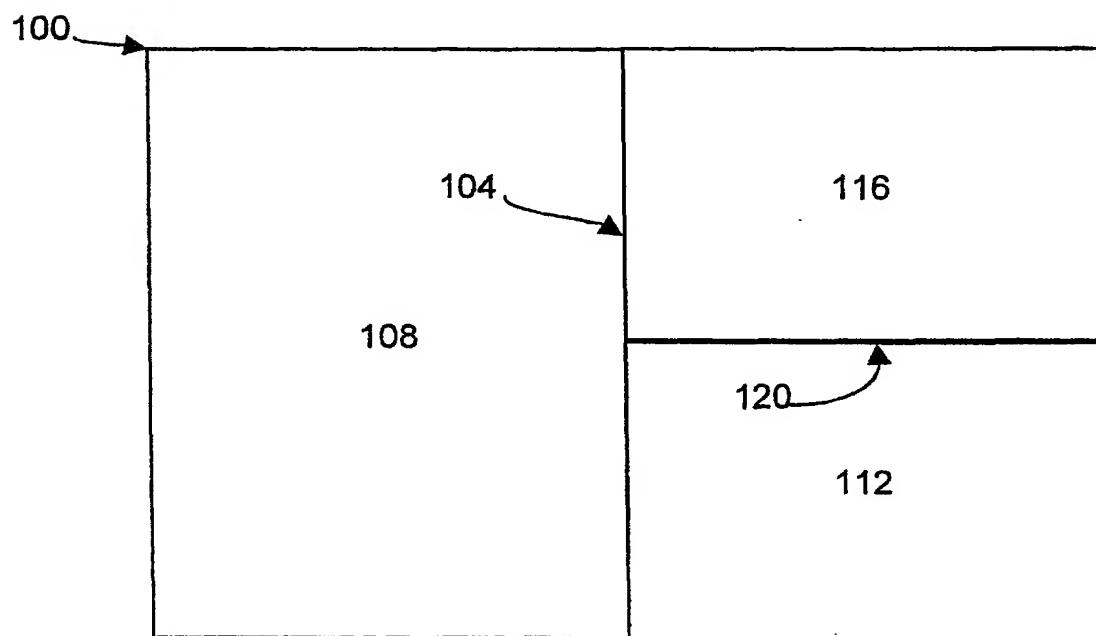


Fig. 4

3/9

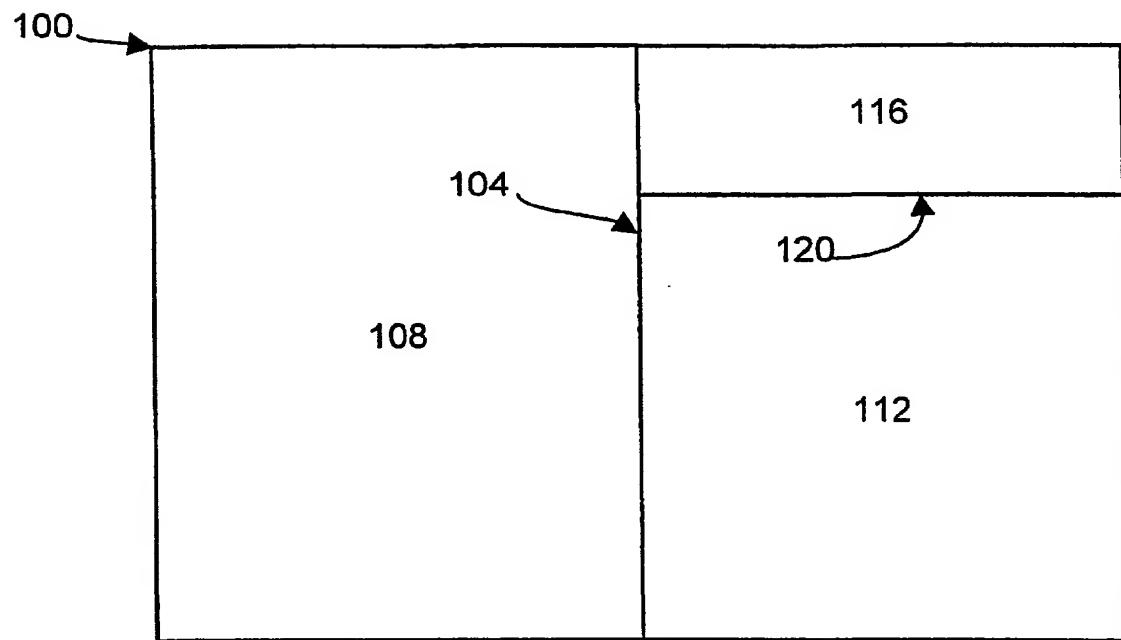


Fig. 5

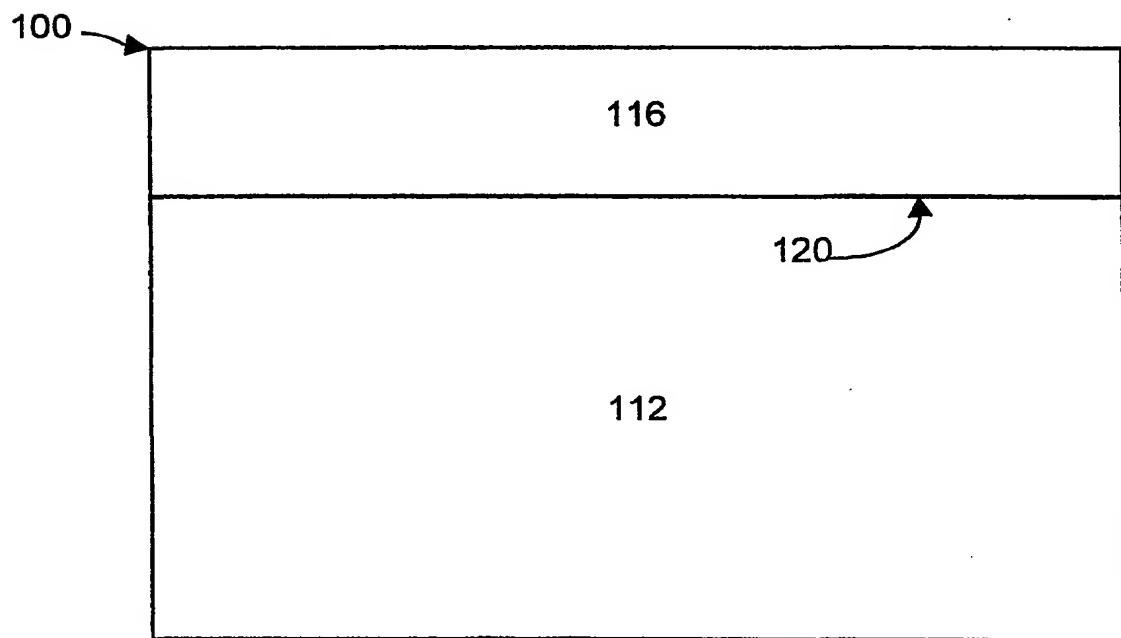


Fig. 6

4/9

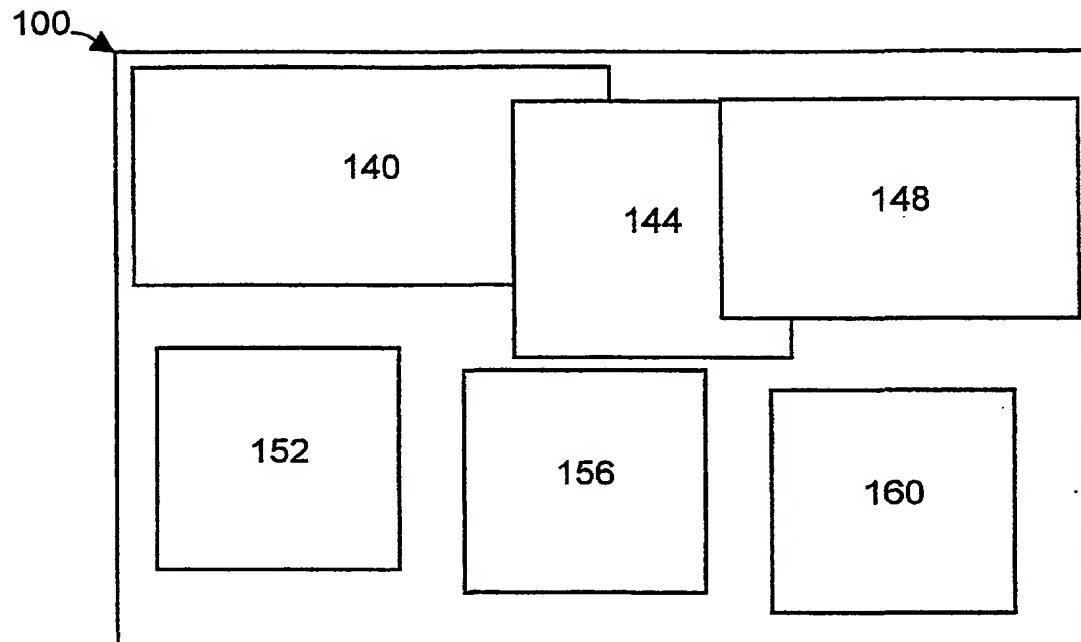


Fig. 7

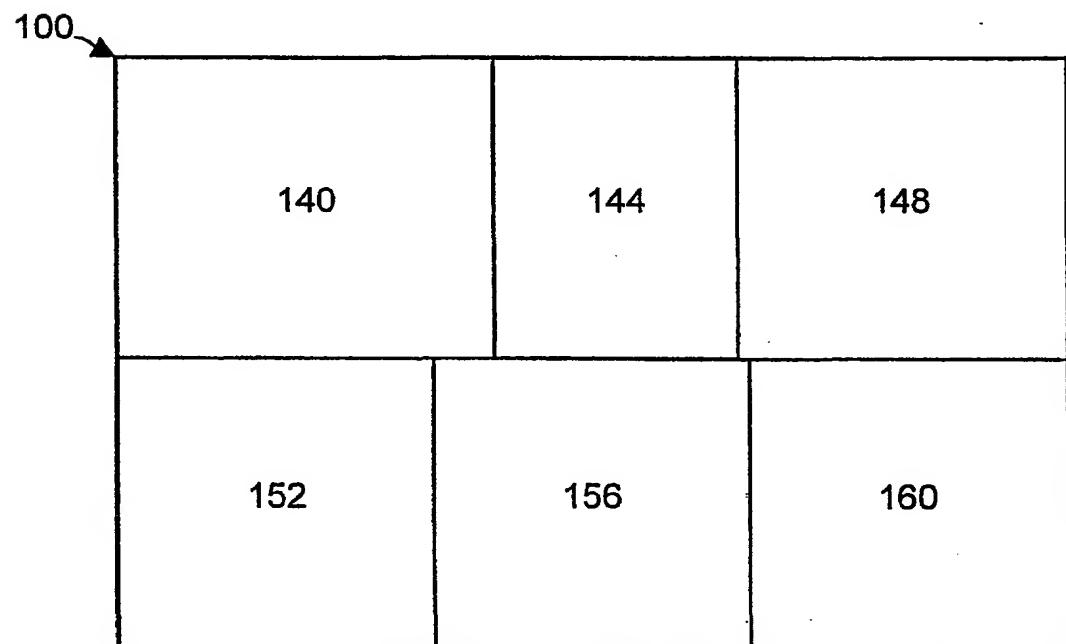


Fig. 8

5/9

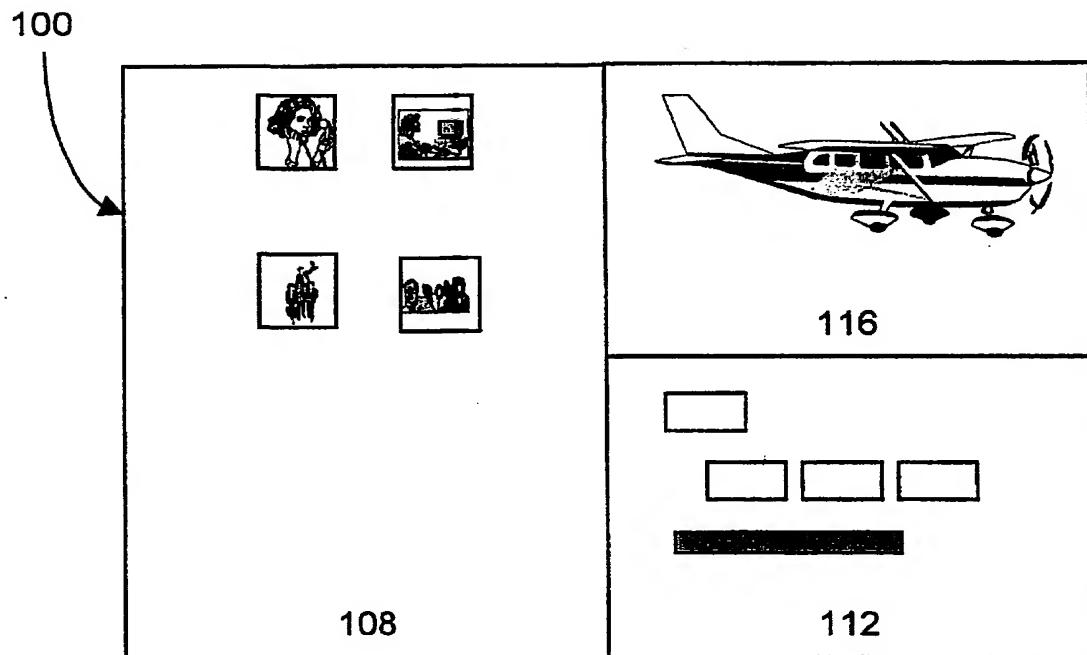


Fig. 9

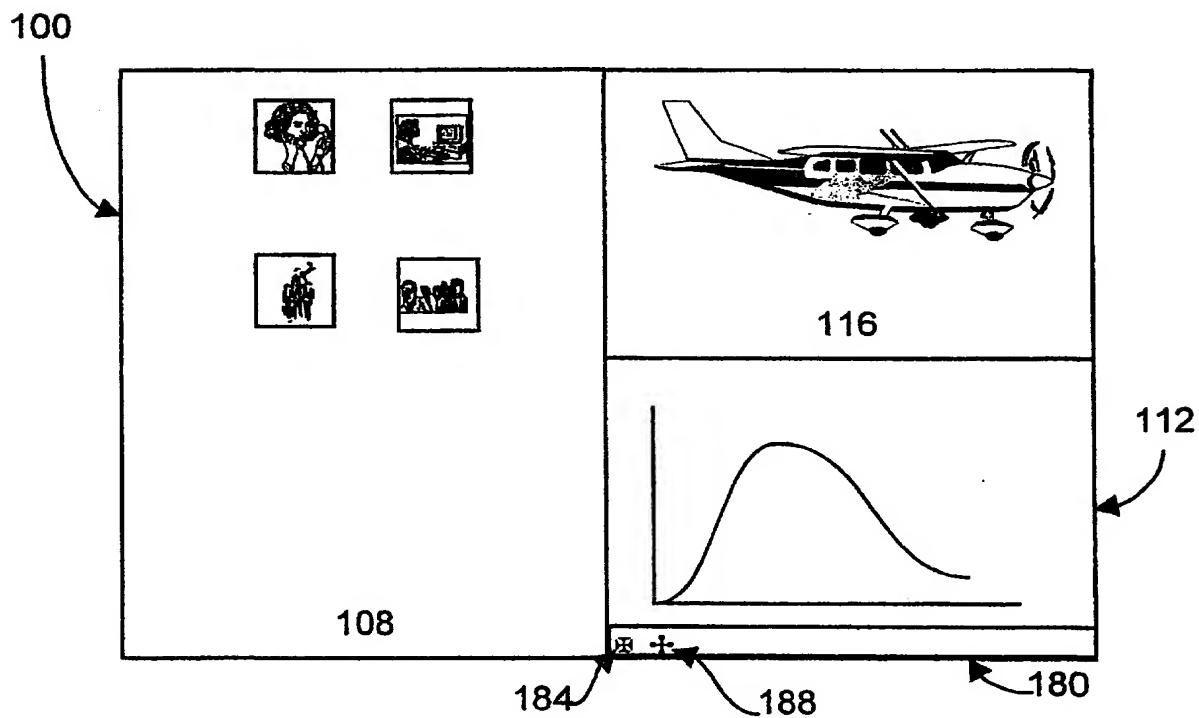


Fig. 10

6/9

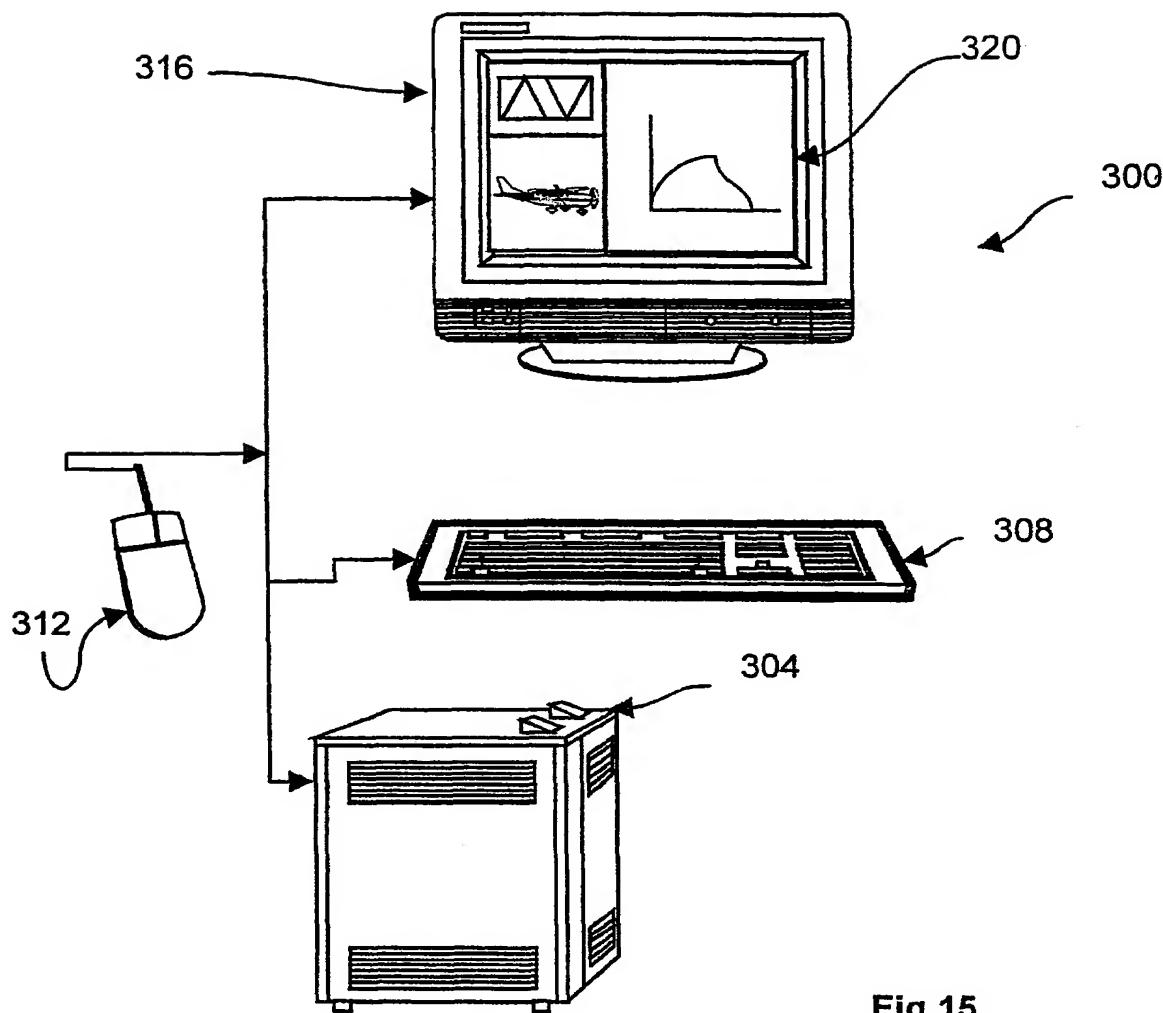


Fig.15

7/9

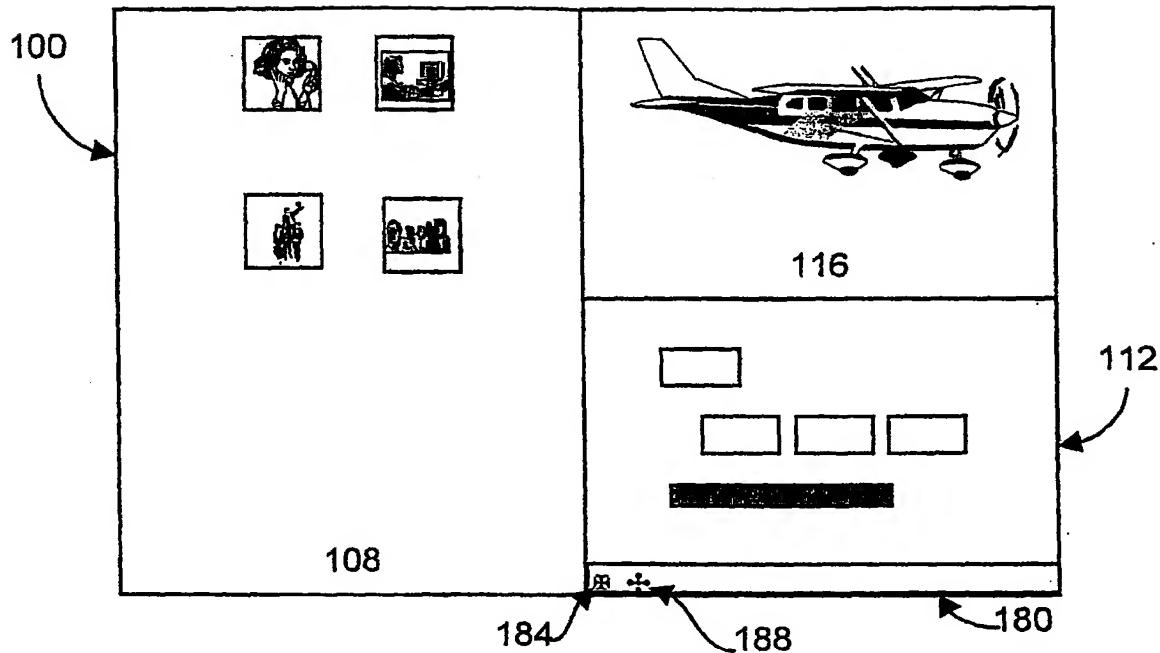


Fig. 11

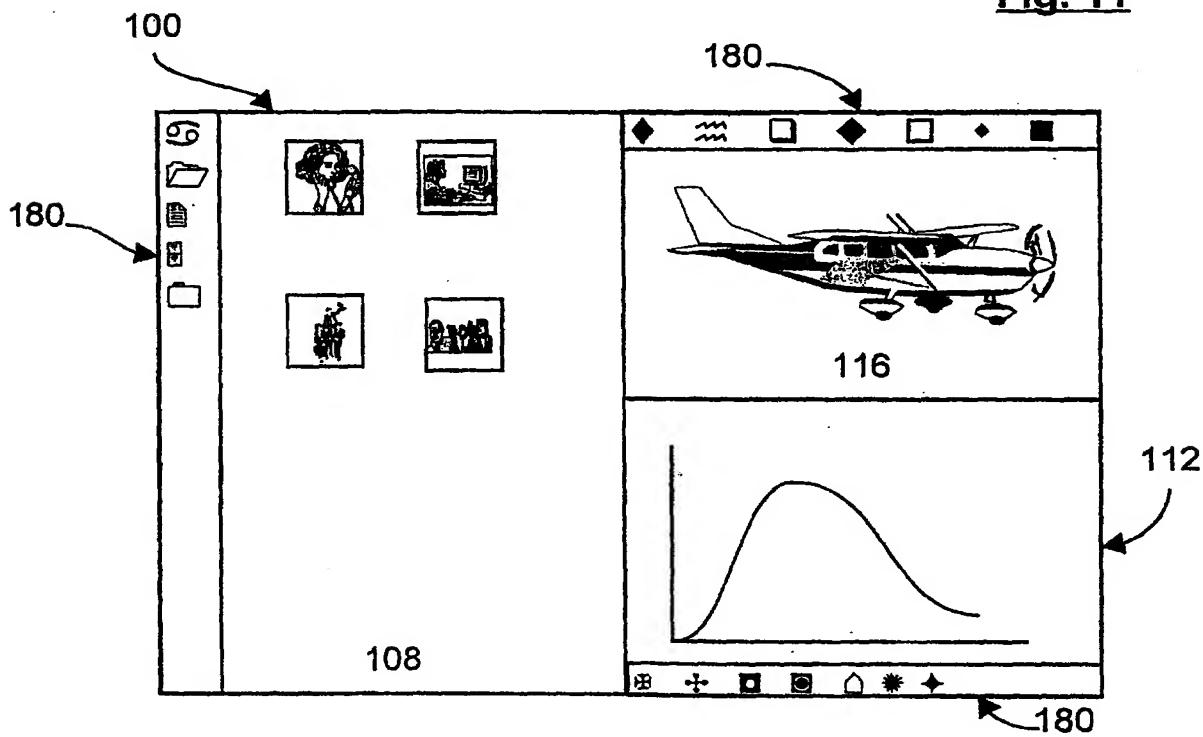


Fig. 12

8/9

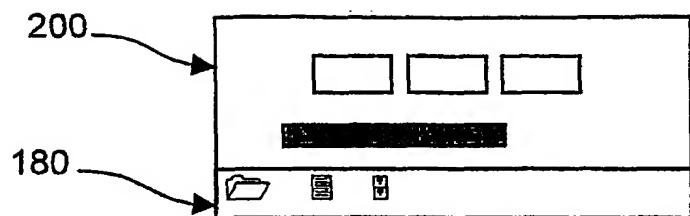


Fig. 13a

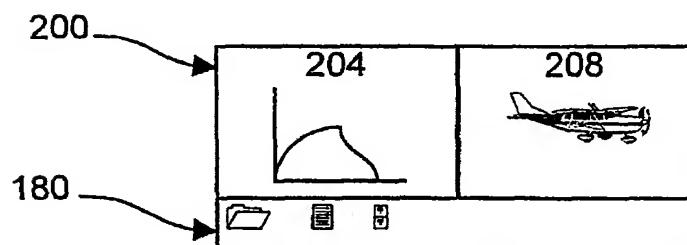


Fig. 13b

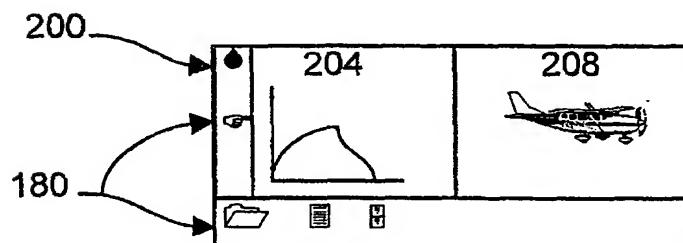


Fig. 13c

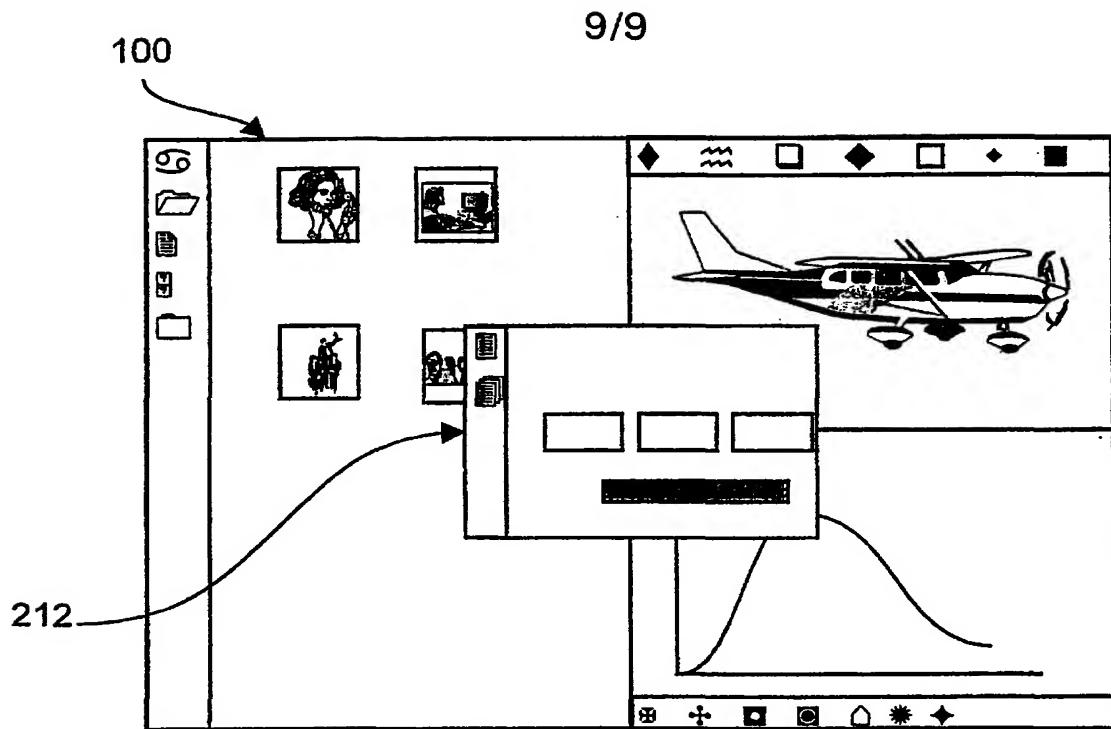


Fig. 14

INTERNATIONAL SEARCH REPORT

Intern. Appl. No.

PCT/CA 00/00393

A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 G06F3/033

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 712 995 A (COHN ROBERT M) 27 January 1998 (1998-01-27) column 10, line 56 -column 14, line 18 column 34, line 57 -column 36, line 7 figures 7A-8,17,18,20A-24F	1,11
A	---	2-9
X	EP 0 740 246 A (CANON KK) 30 October 1996 (1996-10-30) column 7, line 47 -column 14, line 55 figures 2-4,6	1,11
A	-----	2,3,5-7, 9



Further documents are listed in the continuation of box C.



Patent family members are listed in annex.

° Special categories of cited documents :

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier document but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search

13 September 2000

Date of mailing of the international search report

20/09/2000

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl.
Fax: (+31-70) 340-3016

Authorized officer

Baldan, M

INTERNATIONAL SEARCH REPORT

Intern. Appl. Application No

PCT/CA 00/00393

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F3/033

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 712 995 A (COHN ROBERT M) 27 January 1998 (1998-01-27) column 10, line 56 -column 14, line 18 column 34, line 57 -column 36, line 7 figures 7A-8,17,18,20A-24F	1,11
A	---	2-9
X	EP 0 740 246 A (CANON KK) 30 October 1996 (1996-10-30) column 7, line 47 -column 14, line 55 figures 2-4,6	1,11
A	-----	2,3,5-7, 9

Further documents are listed in the continuation of box C.

Patent family members are listed in annex.

* Special categories of cited documents :

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier document but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

13 September 2000

Date of mailing of the international search report

20/09/2000

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Baldan, M